

IMPLEMENTASI MIKROKONTROLER SEBAGAI PENCACAH FREKUENSI BERBASIS PENGUKURAN PERIODE ISYARAT MASUKAN

Freddy Kurniawan

Sekolah Tinggi Teknologi Adisutjipto (STTA)
Jl. Janti, Blok R, Kompleks Lanud Adisutjipto, Yogyakarta
Telpon (0274) 451262, Faks. (0274) 451265
email: freddykurniawan@yahoo.com

Abstract

One of the main components from frequency counter is a processor. This processor must have ability to conduct counting and arithmetical calculation. To get competitive price, in this research a microcontroller is applied as a processor. This paper presents design of frequency counter based on microcontroller AT89S51 which has two independent timers. The input frequency counter is conducted by using facilities inside microcontroller e.g. T0 and external interruption. Counter method used in this research is by measuring input signal period. This method is proven effectively to count frequency especially low frequency. As the result, the frequency counter has successfully measured frequency from 0.1 until 10 kHz with maximum error 1%. This device is possible to be the next frequency counter with competitive price.

Keywords: *microcontroller, AT89S51, frequency counter*

Abstrak

Salah satu komponen utama dari pencacah frekuensi adalah sebuah prosesor. Prosesor tersebut harus dapat melakukan proses pencacahan dan kemudian melakukan operasi aritmatika. Agar harga instrumen pencacah frekuensi menjadi lebih kompetitif, maka pada penelitian ini digunakan sebuah mikrokontroler sebagai pengganti prosesor. Paper ini menghadirkan rancangan pencacah frekuensi berbasis AT89S51 yang memiliki dua buah pewaktu yang tidak saling gayut. Pencacahan frekuensi masukan dilakukan dengan memanfaatkan fasilitas T0 dan interupsi eksternal yang dimiliki oleh mikrokontroler tersebut. Metode pencacah yang digunakan adalah dengan mengukur periode isyarat masukan. Metode ini efektif untuk menghitung frekuensi, terutama frekuensi rendah. Hasil penelitian menunjukkan bahwa pencacah frekuensi yang dirancang dapat digunakan untuk mengukur frekuensi dari 0.1-10kHz dengan galat maksimal 1%. Selanjutnya, peranti ini dapat menjadi alternatif dikembangkannya alat pencacah frekuensi dengan harga yang lebih kompetitif.

Kata kunci: *mikrokontroler, AT89S51, pencacah frekuensi*

1. PENDAHULUAN

Salah satu parameter isyarat elektronis yang sering digunakan adalah besaran frekuensi. Nilai frekuensi dapat dihitung dari nilai periodenya. Satu periode isyarat analog terdiri dari satu lembah dan satu gunung. Sedangkan satu periode isyarat digital terdiri dari satu kali logika rendah dan satu kali logika tinggi. Untuk dapat menentukan nilai frekuensi suatu isyarat dapat digunakan sebuah osiloskop atau sebuah pencacah frekuensi.

Pencacah frekuensi dapat menghitung frekuensi masukan hingga ketelitian tertentu. Pencacah frekuensi analog biasanya dapat menghitung dengan ketelitian hingga dua digit, sementara itu pencacah frekuensi digital dapat menghitung dengan ketelitian dua hingga tiga digit.

Terdapat dua metode yang dapat digunakan untuk menghitung frekuensi. Pertama adalah dengan mencacah frekuensi masukan. Dalam durasi waktu tertentu, banyaknya periode isyarat masukan dicacah. Hasil cacahan dibagi dengan durasi waktu tersebut akan

menghasilkan nilai frekuensi masukan. Sedangkan metode kedua adalah dengan pengukuran periode isyarat masukan. Pengukuran dilakukan atas dua titik bersesuaian yang berurutan. Untuk isyarat clock, pengukuran periode isyarat clock dapat dilakukan dengan mengukur jeda waktu terjadinya dua sisi turun yang berurutan atau mengukur jeda waktu terjadinya dua sisi naik yang berurutan [1].

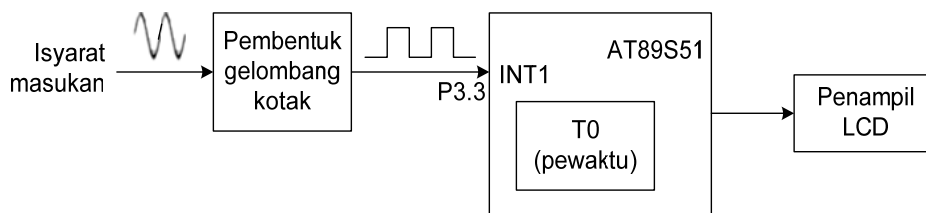
Salah satu komponen utama dari pencacah frekuensi adalah sebuah prosesor. Prosesor tersebut harus dapat melakukan proses pencacahan dan kemudian melakukan operasi aritmatika. Agar harga instrumen pencacah frekuensi menjadi lebih kompetitif, maka dapat digunakan sebuah mikrokontroler sebagai pengganti prosesor [2] [3].

Mikrokontroler keluarga MCS-51 dapat digunakan sebagai pembangkit frekuensi dengan galat di bawah 1 % untuk frekuensi di bawah 10 kHz [4]. Berdasar fakta tersebut, mikrokontroler keluarga MCS-51 ini tentu dapat digunakan sebagai pencacah frekuensi dengan galat cukup rendah untuk kisaran frekuensi tertentu. Salah satu fasilitas yang dimiliki mikrokontroler 8051 yang digunakan dalam proses pengukuran periode isyarat masukan adalah pewaktu (timer). Mikrokontroler seri 8051 mempunyai dua buah pewaktu, yaitu T0 dan T1. Kedua pewaktu tersebut tidak saling gayut (independen). Bahkan T0 dan T1 dapat berjalan bersamaan dengan jalannya program utama. Kedua pewaktu tersebut dapat dioperasikan sebagai pewaktu dengan beberapa mode [5] [6]. Mikrokontroler keluarga 8051 ini juga mempunyai fasilitas interupsi.

Pada penelitian ini dibuat sebuah pencacah frekuensi berbasis mikrokontroler AT89S51. Pencacahan frekuensi masukan dilakukan dengan menggunakan salah satu fasilitas pewaktu dan interupsi eksternal. Metode yang digunakan adalah mengukur periode isyarat masukan. Metode ini efektif untuk menghitung frekuensi, terutama frekuensi rendah. Frekuensi hasil hitungan ditampilkan dalam beberapa digit desimal. Sementara itu, mikrokontroler ini hanya mampu menyimpan dan mengolah secara langsung data 8 bit. Sehingga untuk beberapa proses aritmatika yang melibatkan data hingga 32 bit, harus digunakan beberapa algoritma khusus dengan memanfaatkan memori yang dapat digunakan secara leluasa yang ukurannya tidak lebih dari 64 byte [6].

2. METODOLOGI PENELITIAN

Mikrokontroler dioperasikan dengan frekuensi clock 12 MHz. T0 difungsikan sebagai pewaktu mode 1, sehingga T0 bertindak sebagai pewaktu 16 bit. Batas atas nilai TH0 TL0 adalah $FFFF_{\text{heks}}$ atau 65.535. Setiap diaktifkan, nilai TH0 TL0 direset terlebih dahulu. Sehingga setiap pewaktu T0 diaktifkan, T0 akan mencacah dengan menaikkan TH0 TL0 setiap satu mikrodetik satu nilai, dari 0000h hingga $FFFF_{\text{heks}}$. Pada saat TH0 TL0 mencapai nilai $FFFF_{\text{heks}}$, maka satu mikrodetik kemudian nilai TH0 TL0 kembali ke 0000_{heks}. Pada saat itu terjadi limpahan (overflow) pada T0. Durasi waktu dari dimulainya proses pencacahan hingga terjadi limpahan adalah 10000_{heks} atau 65.536 mikrodetik. Ini merupakan durasi waktu cacahan maksimal T0 atau t_{T0} . Dengan cara ini, T0 dapat digunakan untuk mengukur durasi waktu hingga 65.536 mikrodetik.

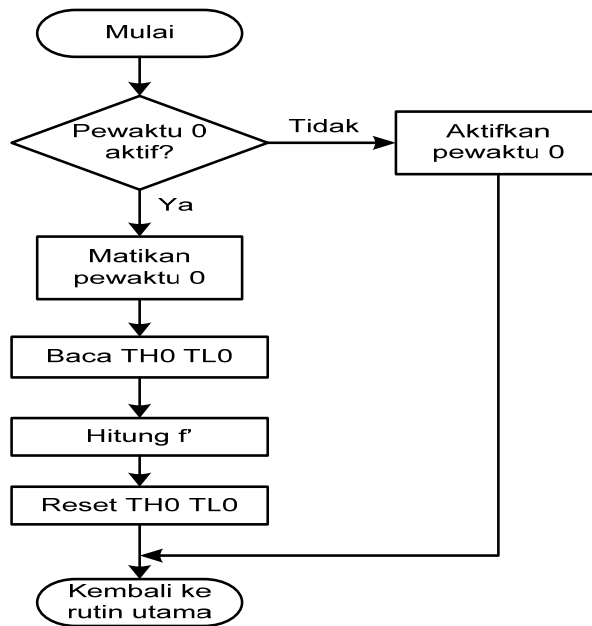


Gambar 1. Diagram blok pencacah frekuensi.

Isyarat masukan dimasukkan ke port 3 pin 3 sebagaimana diagram blok pada Gambar 1. Pada masukan ditambahkan pembentuk gelombang kotak dari gerbang penyangga dengan pemacu Schmitt untuk mengantisipasi adanya isyarat masukan yang belum berupa gelombang kotak. Rutin program utama hanyalah digunakan untuk memberikan tampilan nilai

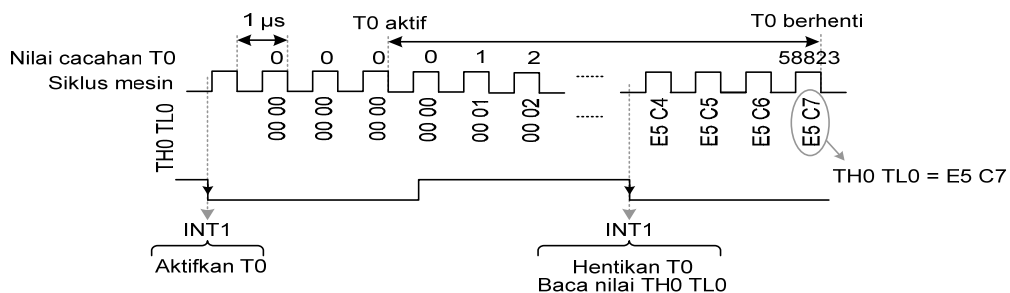
frekuensi pada penampil LCD 1X16 karakter. Nilai frekuensi yang tertampil di LCD cukup diperbarui setiap 0,5 detik.

Dengan mengaktifkan interupsi eksternal 1 (INT1) bertipe sisi (edge), maka pada saat terjadi sisi turun isyarat masukan, interupsi eksternal 1 akan aktif. Sehingga rutin interupsi (ISR) T1 yang berada di alamat 0013h akan dieksekusi. Di lokasi tersebut diberikan instruksi untuk mengaktifkan T0, yaitu dengan mengeset TR0. Sebelum T0 diaktifkan, terlebih dahulu dideteksi aktif tidaknya T0 dengan mengecek nilai TR0. Diagram alir subrutin program layanan interupsi atas aktifnya interupsi eksternal 1 ini dapat dilihat pada Gambar 2.



Gambar 2. Diagram alir subrutin program layanan interupsi eksternal 1

Selang waktu terjadinya interupsi eksternal 1 (INT1) hingga selesainya perintah untuk mengaktifkan T0 adalah 3 mikrodetik. Sehingga cacahan oleh T0 dimulai 3 mikrodetik setelah ada sisi turun isyarat masukan ke kaki INT1 sebagaimana Gambar 3. Pada saat terjadi sisi turun isyarat masukan berikutnya, atau dengan kata lain isyarat masukan sudah berlangsung satu periode, INT1 kembali aktif. Aktifnya INT1 untuk yang kedua kali akan menghentikan T0. Di sini, dicek kembali nilai TR0. Jika TR0 = 1, berarti T0 sedang aktif, program akan menghentikan T0. Selang waktu terjadinya INT1 yang kedua kali hingga berhentinya T0 adalah 3 mikrodetik. Nilai akhir TH0 TLO merupakan hasil pengukuran periode isyarat masukan dengan satuan mikrodetik. Gambar 3 merupakan grafik cacahan T0 atas isyarat masukan dengan frekuensi 17 Hz.



Gambar 3. Cacahan atas isyarat masukan berfrekuensi 17 Hz.

Pada penelitian ini, T0 digunakan untuk menghitung durasi waktu satu periode isyarat masukan. Nilai frekuensi masukan dapat dihitung dengan persamaan (1).

$$f = \frac{1}{T} \quad (1)$$

Satu cacahan membutuhkan waktu 1 mikrodetik. Sehingga secara teori nilai cacahan TH0 TL0 untuk frekuensi masukan f dapat ditentukan menggunakan Persamaan (2).

$$\text{TH0 TL0} = \frac{10^6}{f} \quad (2)$$

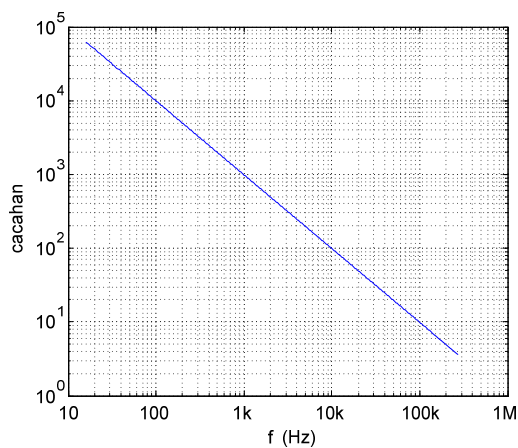
Nilai TH0 TL0 selalu berbentuk bilangan bulat. Di sini terjadi pembulatan. Frekuensi hasil hitungan dapat ditentukan menggunakan Persamaan (3).

$$f = \frac{10^6}{\text{TH0TL0}} = \frac{F4240_{\text{heks}}}{\text{TH0TL0}} \quad (3)$$

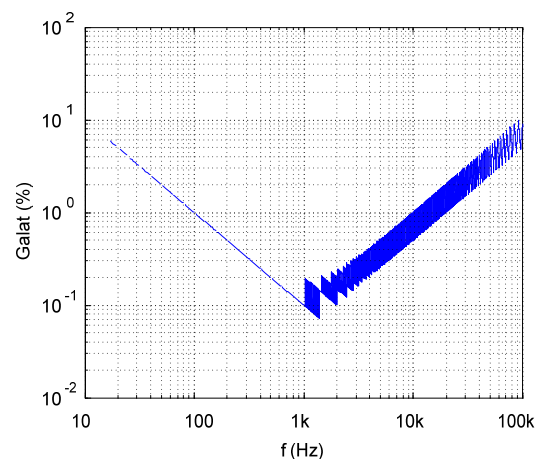
3. HASIL DAN PEMBAHASAN

Semakin tinggi frekuensi isyarat masukan, semakin rendah pula nilai cacahan T0 yang tersimpan di register TH0 dan TL0. Perubahan nilai cacahan atas berubahnya frekuensi isyarat masukan dapat dilihat pada Gambar 4.

Isyarat berfrekuensi 16 Hz akan dicacah 62.500 kali. Sementara itu, menurut teori isyarat berfrekuensi yang masih dapat dicacah dengan benar adalah 1 MHz. Frekuensi dengan periode 1 mikrodetik tersebut akan dicacah 1 kali. Namun dalam kenyataan aplikasi ini hanya sanggup mencacah dengan benar frekuensi di bawah 167 kHz.



Gambar 4. Cacahan di TH0 TL0 untuk $16 \text{ Hz} \leq f \leq 167 \text{ kHz}$



Gambar 5. Galat terbesar untuk $16 \text{ Hz} \leq f \leq 100 \text{ kHz}$

Pada aplikasi ini digunakan interupsi berjenis sisi (*edge*). Dengan digunakannya interupsi berjenis sisi, adanya sisi turun isyarat di INT1 akan diabaikan selama subrutin program layanan interupsi belum selesai dikerjakan. Dengan kata lain, selama mikrokontroler masih mengerjakan subrutin layanan interupsi eksternal 1 sebagaimana diagram alir pada Gambar 2, maka interupsi eksternal 1 baru akan diabaikan. Dalam kenyataan, subrutin untuk mengecek aktif tidaknya pewaktu 0, mengaktifkan pewaktu 0, dan kemudian kembali ke program utama (*return interrupt*) memerlukan waktu 5 mikrodetik. Sehingga selang waktu terjadinya dua

interupsi ekstremal berurutan minimal adalah 6 mikrodetik. Akibatnya, aplikasi ini hanya dapat mencacah dengan benar isyarat dengan periode minimal 6 mikrodetik. Sehingga frekuensi tertinggi isyarat yang dapat dicacah dengan benar adalah 166.666 Hz.

Frekuensi isyarat terendah yang dapat diukur oleh aplikasi ini adalah 15,3 Hz. Isyarat tersebut mempunyai periode 65.359 mikrodetik. Hal ini disebabkan nilai maksimal cacahan TH0 TL0 yang tersimpan di TH0 TL0 adalah FFFF_{heks} atau 65535_{des}, dengan waktu cacahan hingga terjadi limpahan adalah 65.536 mikrodetik. Ini berarti, secara teori frekuensi minimal isyarat yang dapat diukur adalah 15,29 Hz.

3.1. Ketepatan Hasil Hitungan

Tidak semua nilai frekuensi isyarat masukan akan dicacah dengan tepat. Sebagai contoh, sesuai Persamaan (2), isyarat berfrekuensi 17 Hz seharusnya dicacah 58.823,53 kali. Padahal nilai cacahan yang disimpan dalam TH0 TL0 selalu berbentuk bilangan bulat. Hasil cacahan atas isyarat tersebut dapat mempunyai dua kemungkinan. Kemungkinan pertama, isyarat tersebut akan dicacah 58.823 kali dengan TH0 TL0 = E5C7_{heks}, kemungkinan kedua isyarat tersebut akan dicacah 58.824 kali dengan TH0 TL0 = E5C8_{heks}.

Sesuai Persamaan (3), hasil cacahan pertama akan menghasilkan $f' = 17,0002$ Hz. sedangkan hasil cacahan kedua akan menghasilkan $f' = 16,9999$ Hz. Namun karena pembagian pada mikrokontroler selalu menghasilkan bilangan bulat, hasil bagi 10^6 atau F4240_{heks} dengan E5C7_{heks} akan menghasilkan nilai $f' = 11$ _{heks} atau 17_{des} Hz. Sedangkan dari hasil cacahan kedua akan dihasilkan nilai $f' = 10$ _{heks} atau 16_{des} Hz.

Galat atas frekuensi hasil hitungan dapat ditentukan dengan Persamaan (4).

$$\text{Galat} = \frac{|(f - f')|}{f} \times 100\% \quad (4)$$

Nilai f' pertama pada contoh di muka mempunyai galat $\frac{|(16 - 17)|}{17} \times 100\% = 5,9\%$, sedangkan nilai f' kedua mempunyai galat 0%. Sehingga galat terbesar untuk $f = 17$ Hz adalah 5,9%. Nilai galat terbesar yang dapat terjadi akan semakin menurun dengan naiknya frekuensi masukan. Nilai galat terkecil ada di sekitar frekuensi 1 kHz.

Untuk frekuensi masukan di atas 1 kHz, nilai galat justru akan semakin tinggi sebagaimana Gambar 5. Nilai galat menjadi sangat berfluktuatif untuk kisaran frekuensi ini. Sebagai contoh, sesuai Persamaan (2), isyarat berfrekuensi 99 kHz akan dicacah 10,101 kali. Dalam satu periode, nilai cacahan yang tersimpan di TH0 TL0 dapat bernilai 00 0A_{heks} atau 00 0B_{heks}. Nilai pertama akan menghasilkan $f' = 100$ kHz, sedangkan nilai kedua akan menghasilkan $f' = 90,9$ kHz. Nilai f' kedua mempunyai galat 8,17%.

3.2. Koreksi Galat untuk $f < 100$ Hz

Sebagaimana Gambar 5, nilai hitungan atas frekuensi masukan di bawah 100 Hz dapat memberikan galat di atas 1%. Untuk menurunkan nilai galat, maka nilai pembilang pada Persamaan (3) dapat dinaikkan. Agar proses penskalaan menjadi lebih mudah, maka nilai pembilang dikalikan 10 untuk $10 \text{ Hz} \leq f < 100 \text{ Hz}$.

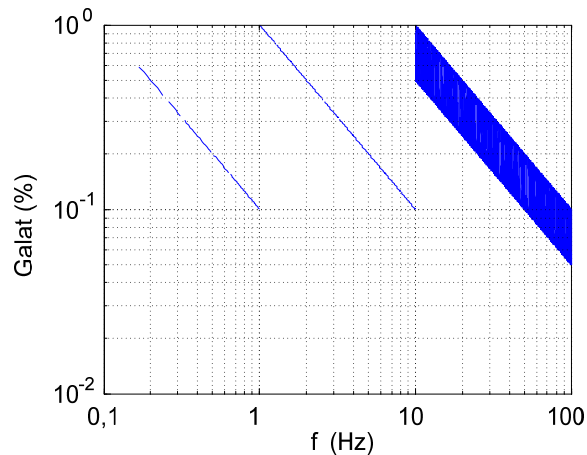
$$f' = \frac{10^7}{\text{TH0TL0}}, \text{ untuk } 10 \text{ Hz} \leq f < 100 \text{ Hz} \quad (5)$$

Pada Persamaan (5) tersebut, nilai f' menjadi 10 kali f . Ini berarti jika diperoleh nilai $f' = \text{XXX}$, maka nilai tersebut harus ditampilkan sebagai XX,X Hz. Dengan cara yang sama, galat untuk masukan dengan frekuensi di bawah 10 Hz pun dapat dikoreksi. Perhitungan untuk mendapatkan f' untuk $f < 10$ Hz mengikuti Persamaan (6) dan (7).

$$f' = \frac{10^8}{\text{TH0TL0}}, \text{ untuk } 1 \text{ Hz} \leq f < 10 \text{ Hz} \quad (5)$$

$$f' = \frac{10^9}{TH0TL0}, \text{ untuk } 0,1 \text{ Hz} \leq f < 1 \text{ Hz} \quad (6)$$

Pada Persamaan (6), nilai f' menjadi 100 kali f . Ini berarti jika diperoleh nilai $f' = XXX$, maka nilai tersebut harus ditampilkan sebagai X,XX Hz. Sedangkan pada Persamaan (7) tersebut, nilai f' adalah 1000 kali f . Ini berarti jika diperoleh nilai $f' = XXX$, maka nilai tersebut harus ditampilkan sebagai $0,XXX$ Hz. Dengan cara tersebut, galat maksimal hasil hitungan untuk frekuensi masukan di bawah 100 Hz yang tadinya di atas 1 % sebagaimana Gambar 5 dapat diturunkan menjadi di bawah 1 % sebagaimana Gambar 6.



Gambar 6. Galat maksimal untuk $0,1 \text{ Hz} \leq f \leq 100 \text{ Hz}$

3.3. Pencacahan Frekuensi Masukan di Bawah 15,29 Hz

Isyarat masukan dengan frekuensi di bawah 15,29 Hz mempunyai periode lebih dari 65.536 mikrodetik. Dalam durasi waktu tersebut, pada hasil cacahan TH0 TL0 sudah terjadi limpahan. Untuk pengukuran frekuensi yang lebih rendah lagi, limpahan dari TH0 TL0 dapat terjadi lebih dari satu kali. Ini berarti harus digunakan suatu penampung yang akan mencacah jumlah terjadinya limpahan. Sehingga Persamaan (5), (6) dan (7) tidak dapat langsung digunakan.

Setiap kali terjadi limpahan, isi sebuah memori 8 bit bernama jumlah_limpahan dinaikkan satu. Setelah T0 berhenti, nilai cacahan_total dihitung dengan menggunakan Persamaan (8).

$$\text{cacahan_total} = \text{jumlah_limpahan} \times 10000_{\text{heks}} + \text{TH0 TL0} \quad (8)$$

Dengan menetapkan frekuensi minimal yang dapat diukur adalah 0,1 Hz, maka nilai jumlah_limpahan maksimal adalah 152 atau 98_{heks} . Hasil kali jumlah_limpahan dengan 10000_{heks} pada Persamaan (8) merupakan variabel 24 bit dengan format $XX0000_{\text{heks}}$.

$$f = \begin{cases} \frac{3B9ACA00_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 0,1\text{Hz} \leq f < 1\text{Hz} \\ \frac{5F5E100_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 1\text{Hz} \leq f < 10 \text{ Hz} \\ \frac{989680_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 10 \text{ Hz} \leq f < 100 \text{ Hz} \\ \frac{F4240_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 100 \text{ Hz} \leq f < 1\text{MHz} \end{cases} \quad (9)$$

Nilai hasil kali tersebut didapat dengan cara memasukkan nilai jumlah_limpahan ke byte paling kiri cacahan_total. Sedangkan penambahan dengan TH0 TL0 cukup dilakukan dengan masukkan nilai TH0 TL0 ke byte tengah dan bit paling kanan variabel tersebut. Variabel cacahan_total mempunyai panjang 24 bit.

Dengan menggabungkan Persamaan (5) hingga Persamaan (8) dan mengubah semua nilai ke format heksadesimal, nilai frekuensi masukan f' dapat' dihitung menggunakan Persamaan (9). Dalam pemrograman, seleksi atas satu di antara empat persamaan pada Persamaan (9) yang digunakan dilakukan atas nilai cacahan_total, bukan atas frekuensi masukan f yang belum diketahui. Sehingga Persamaan (8) diubah menjadi Persamaan (9).

$$f' = \begin{cases} \frac{3B9ACA00_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 989680_{\text{heks}} \leq \text{cacahan_total} < F4240_{\text{heks}} \\ \frac{5F5E100_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } F4240_{\text{heks}} \leq \text{cacahan_total} < 186A0_{\text{heks}} \\ \frac{989680_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 186A0_{\text{heks}} \leq \text{cacahan_total} < 2710_{\text{heks}} \\ \frac{F4240_{\text{heks}}}{\text{cacahan_total}}, & \text{untuk } 2710_{\text{heks}} \leq \text{cacahan_total} < 0_{\text{heks}} \end{cases} \quad (10)$$

Pada Persamaan (10) tersebut, pembilang menempati data 32 bit, sedangkan penyebut menempati data 24 bit. Meskipun disediakan memori dengan alokasi 24 bit untuk variabel cacahan_total, namun variabel tersebut tidak selalu menggunakan alokasi memori sebesar 24 bit. Dengan panjang data cacahan_total yang mencapai 24 bit, cukup menyulitkan jika cacahan_total diperlakukan sebagai pembagi. Untuk mengurangi tingkat kerumitan proses pembagian, dilakukan pembulatan dengan cara menyederhanakan pembilang dan penyebutnya. Pembagian dilakukan oleh data cacahan_total dengan panjang 16 bit terhadap data pembilang dengan panjang 24 bit dengan suatu algoritma khusus.

Jika byte paling kiri variabel cacahan_total tidak sama dengan nol, maka pembilang dan penyebut dapat dibagi dua. Pembagian dengan dua atas suatu nilai cukup dilakukan dengan menggeser bit-bit yang membentuk nilai tersebut ke kanan sejauh satu bit. Jika byte paling kiri variabel cacahan_total belum sama dengan nol, maka proses tersebut dapat diulang. Cara tersebut efektif untuk melakukan penyederhaan atas pencacahan untuk frekuensi rendah. Penyederhaan atas pembilang dan penyebut tidak akan banyak mempengaruhi nilai galat. Cara ini efektif untuk nilai frekuensi masukan hingga 0,1 Hz.

Pada isyarat masukan dengan frekuensi 0,1 Hz, maka cacahan_total bernilai 10 juta atau 989680_{heks} . Nilai cacahan_total = 989680_{heks} . Karena byte paling kiri cacahan_total = 98, maka pembilang dan penyebut dibagi dua. Proses pembagian dengan dua dilakukan hingga cacahan_total = 9896_{heks} . Nilai pembilang sekarang adalah $3B9ACA_{\text{heks}}$. Sehingga didapat nilai frekuensi hasil hitungan

$$f = \frac{3B9ACA00_{\text{heks}}}{989680_{\text{heks}}} \cong \frac{3B9ACA_{\text{heks}}}{9896_{\text{heks}}} \cong 64_{\text{heks}} = 100_{\text{des}} \quad (11)$$

Pembagian dengan 9896_{heks} atas $3B9ACA_{\text{heks}}$ akan menghasilkan nilai 100. Nilai tersebut akan ditampilkan sebagai 0,1 Hz. Inilah nilai frekuensi terkecil yang dapat ditampilkan sistem.

3.4. Analisis Ketepatan Frekuensi di Atas 100 Hz

Pada Gambar 5 terlihat bahwa untuk kisaran $100 \text{ Hz} \leq f < 10 \text{ kHz}$, galat yang terjadi maksimal 1 %. Pada kisaran tersebut dapat digunakan ketepatan hingga 2 digit. Format tampilan frekuensi hasil hitungan yang adalah XX0 Hz untuk kisaran $100 \text{ Hz} \leq f < 1 \text{ kHz}$, dan X,X kHz untuk kisaran $1 \leq f < 10 \text{ kHz}$.

Pada kisaran $10 \text{ kHz} \leq f < 100 \text{ kHz}$, galat maksimal adalah 10 %. Pada kisaran tersebut dapat digunakan ketepatan 1 digit. Format tampilan frekuensi hasil hitungan adalah X0 kHz. Sedangkan pada kisaran $f \geq 100 \text{ kHz}$, galat yang terjadi sudah sedemikian besar. Frekuensi hasil hitungan tidak layak lagi untuk digunakan.

Sementara itu, pada Gambar 6 terlihat bahwa untuk kisaran $0,1 \leq f < 100$ Hz, galat maksimal yang terjadi 1 %. Pada kisaran tersebut dapat digunakan ketepatan hingga 2 digit. Format tampilan frekuensi hasil hitungan adalah XX Hz untuk $10 \leq f < 100$ Hz, X,X Hz untuk $1 \leq f < 10$ Hz, dan 0,XX untuk $0,1 \leq f < 1$ Hz. Galat dan format tampilan nilai frekuensi keluaran dapat dilihat pada Tabel 1.

Tabel 1 Kisaran frekuensi masukan, galat dan format tampilan keluaran

Kisaran frekuensi (Hz)	Galat	Tampilan (Hz)
$0,1 \leq f < 1$	< 1 %	0,XX
$1 \leq f < 10$	< 1 %	X,X
$10 \leq f < 100$	< 1 %	XX
$100 \leq f < 1 \text{ k}$	< 1 %	XX0
$1 \text{ k} \leq f < 10 \text{ k}$	< 1 %	X.X00
$10 \text{ k} \leq f < 100 \text{ k}$	< 10 %	X0.000

4. SIMPULAN

Paper ini telah menghadirkan rancangan pencacah frekuensi berbasis AT89S51. Pencacahan frekuensi masukan dilakukan dengan memanfaatkan fasilitas pewaktu T0 dan interupsi eksternal yang dimiliki oleh mikrokontroler tersebut. Metode yang digunakan adalah mengukur periode isyarat masukan. Hal ini dilakukan dengan cara mengukur durasi waktu antara terjadinya NGT berturut-turut dari sinyal input. Hasil penelitian menunjukkan bahwa pencacah frekuensi yang dirancang dapat digunakan untuk mengukur frekuensi dari 0.1-10 kHz dengan galat maksimal 1% dan dapat digunakan untuk mengukur frekuensi hingga 100 kHz dengan galat di bawah 10 %, sehingga dapat menjadi alternatif dikembangkannya alat pencacah frekuensi dengan harga yang lebih kompetitif.

Daftar Pustaka

- [1] Agilent Technologies, "**Fundamentals of the Electronic Counters**", Application Note 200 Electronic Counter Series, Hewlett-Packard Co., Englewood, 1997.
- [2] Kamal, Ibrahim, "**5 Hz to 500 kHz Frequency-Meter: Home-made, Accurate, and Simple Solution**", http://www.ikalogic.com/freq_meter.php, 2008.
- [3] Matjaz Vidmar: "**A Microwave Analog Frequency Divider**", Microwave Journal, International Edition, ISSN 0192-6225, pp: 120-126, November 1998.
- [4] Kurniawan, Freddy, "**Analisis Ketepatan Frekuensi Pembangkit Clock Berbasis Mikrokontroler**", Jurnal Teknoin, Vol. 2 No., Juli 2008.
- [5] Atmel Corporation, "**Microcontroller Instruction Set**", Atmel Corporation, San Jose, USA., 1997.
- [6] Atmel Corporation, "**Atmel 8051 Microcontrollers Hardware Manual**", Atmel Corporation, San Jose, U.S.A., 2007.